LOG PreProcessing: Web Kullanım Madenciliği Ön İşlem Aşaması Uygulma Yazılımı

Turgut Özseven[1], Muharrem Düğenci[2]
[1] Gaziosmanpaşa Üniversitesi , Turhal Meslek Yüksekokulu, Tokat
[2] Karabük Üniversitesi, Endüstri Mühendisliği Bölümü, Karabük
turgutozseven@gmail.com, mdugenci@gmail.com

Özet: İnternet günümüzde reklam, e-ticaret, bilgi ve belge paylaşımı, bankacılık işlemleri, kurumsal işlemler ve eğitim gibi birçok alanda kullanılmaktadır. İnternet üzerinde bulunan bilgiler kadar arka planda depolanan veriler de önemli bilgiler içermektedir. Bu veriler analiz edilmeden sadece depolandığı sürece veri olarak kalmakta ve bilgiye dönüştürülememektedir. Bu çalışmada, web sunucu üzerinde tutulan erişim kayıtlarına web kullanım madenciliği ön işlem aşamasını uygulayarak verileri daha kolay analiz edilebilir duruma getirmek için "LOG PreProcessing" isminde bir yazılım geliştirilmiştir.

Anahtar Kelimeler: Veri madenciliği, web madenciliği, web kullanım madenciliği

LOG PreProcessing: Pre-Processing Phase of Web Usage Mining Application Software

Abstract: Today, Internet is used in many areas such as advertising, e-commerce, information and document sharing, banking, corporate transactions and education. Not only the informations on the internet but also the data stored in the background inculude important informations. As long as this data is only stored without being analyzed these datas can not be transformed into information. In this study, to make the data can be analyzed more easily developed a software named "LOG PreProcessing"developed software named to access records kept on web server pre-proceesing phase of web usage mining is applying

Keywords: Data mining, web mining, web usage mining

1. Giriş

Teknolojinin gelişmesi ve ucuzlamasıyla birlikte işlem gören ve depolanan veri miktarı her geçen gün artmaktadır. Depolanan veriler anlamlandırılmadan sadece depolandığı sürece sahibi olan kurum veya kuruluş açısından bir anlam ifade etmemekte ve depolama gibi ek problemler oluşturmaktadır. Veri madenciliği sayesinde bu veriler analiz edilerek kurum veya kuruluş için kullanışlı bilgiler elde edilmesi ve karar süreçlerinin kısaltılması sağlanabilir.

İnternet artık günümüzde yaşamımızın her aşamasında kullandığımız önemli bir bilgi kaynağı haline gelmiştir. Aynı şekilde internet de kullanıcılar ve sahibi olan kuruluşlar için önemli bilgiler elde edilmesini sağlayacak ve keşfedilmeyi bekleyen önemli bilgiler içermektedir. Web madenciliği sayesinde internet üzerinde bulunan veya depolanan verilerin veri madenciliği teknikleri ile analiz edilmesi ve önemli bilgilerin keşfedilmesi sağlanabilir. Web madenciliği, web sitelerini ziyaret eden kullanıcıların davranışlarını inceleyerek web sitelerinin güncellenmesi veya geliştirilmesi, müşterilerin ilgi alanları, reklam alma, pazarlama stratejileri oluşturma, sayfa kullanım dağılımlarını belirleme gibi birçok konuda karar verilmesini sağlayan bilgileri sunar.

2. Web Madenciliği

Günümüzde internet başta iletişim olmak üzere e-ticaret, reklam, bilgi ve belge paylaşımı, bankacılık işlemleri, kurumsal işlemler ve eğitim gibi birçok alanda kullanılmaktadır. İnternetin herkese açık olması, içerdiği bilgilerin her geçen gün daha düzensiz olmasına ve daha da artmasına neden olmaktadır. Web ortamındaki bu verilerin büyük olması kadar düzensiz olması da web madenciliğine ayrı bir önem kazandırmaktadır [1].

Web madenciliği ilk olarak 1996 yılında Oren Etzioni tarafından ortaya atılmıştır [2]. Bu bildiride Etzioni'ye göre(1996) web madenciliği, veri madenciliği tekniklerini kullanarak www'de bulunan dosya ve servislerden otomatik olarak bilginin ayıklanması, ortaya çıkartılması ve analiz edilmesidir.

Web madenciliği çalışma alanlarının kapsamlı ve detaylı olması bu alanda düzenli bir sınıflandırmayı da gerektirmektedir. Web madenciliği ilk ortaya atıldığı dönemlerde Web İçerik Madenciliği (Web Content Mining) ve Web Kullanım Madenciliği (Web Usage Mining) olmak üzere iki sınıfa ayrılmaktaydı. Web madenciliğinin yaygınlaşması ile birlikte Web Yapı Madenciliği de (Web Structure Mining) üçüncü bir sınıf olarak eklenmiştir [2, 3].

Web içerik madenciliği, www 'de bulunan içerik verisinden kullanışlı bilgi çıkarım işlemini gerçekleştirir [4]. Web yapı madenciliği, web sayfaları ve web siteleri arasındaki bağlantıları yani web yapı verisini inceleyerek bilgi çıkarım işlemini gerçekleştirir [4,5]. Web log mining olarak da bilinen web kullanım madenciliği ise sunucu üzerinde tutulan kullanıcı erişim kayıt dosyalarından(log) bilgi çıkarım işlemini gerçekleştirir.

3. Web Kullanım Madenciliği

Ziyaretçilerin bir web sitesi üzerinde yapmış olduğu her türlü işlem kayıt altına alınmaktadır. Bu kayıtlar web sunucusuna ait erişim kayıtları, uygulama sunucusuna ait kayıtlar, çerezler ve kullanıcı profillerinden oluşmaktadır. Web kullanım madenciliğinde çoğunlukla web sunucusuna ait erişim kayıtları(log) veri kaynağını oluşturmaktadır [6,7].

Web kullanım madenciliği, ziyaretçinin siteyi kullanırken gerisinde bıraktığı erişim verilerinden bilgi üretmeyi amaçlar. Bu amaçla log dosyalarından en yoğun ve en ilginç kullanıcı erişim örüntülerini keşfetmek ve anlamlı verileri çıkartmak için veri madenciliği tekniklerini kullanır [4].

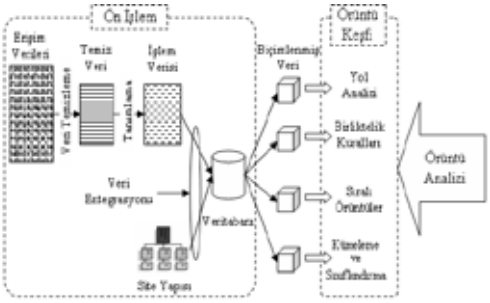
Web kullanım madenciliği ile web yöneticisi için, web sunucusuna gelen taleplerin zamana, kullanıcılara ve URL tiplerine göre dağılımları, başarılı ve başarısız erişimler, gelinen kaynağın belirlenmesi, ziyaretçi tiplerinin belirlenmesi, kurum içi erişim dağılımlarının belirlenmesi, sık ve birlikte ziyaret edilen sayfaların belirlenmesi gibi birçok bilgi sağlanmaktadır. Bu bilgiler yardımıyla web yöneticisi site üzerinde gerekli güncelleştirme ve düzenlemeleri yapabilir, kurum veya kuruluşlar müşterilerine yönelik reklam kampanyaları düzenleyebilir ve ziyaretçilere ürün tavsiyesinde bulunabilir.

Web sitelerinin erişim bilgileri sunucu üzerinde bulunan erişim log dosyalarında tutulmaktadır. Oluşturulan her bir log dosyası sunucu tarafından otomatik olarak oluşturulur ve her gün için ayrı bir log dosyası oluşturulmaktadır. Web sitesine ait alt domainler mevcut ise sunucu tarafından her alt domain için ayrı klasörler oluşturularak erişim bilgileri bu klasörlerde tutulur. Ziyaretçilerin her bir erişimi log dosyasına yeni bir satır olarak eklenir. Eklenen her bir satır erişimle ilgili çeşitli bilgiler tutmaktadır. Tutulan bilgi türleri kullanılan web sunucusuna ve kullanılan log formatına göre farklılık gösterebilir. Ayrıca sunucu üzerinde yapılan ayarlamalara göre tutulacak bilgi türü sayısı artırılabilir veya azaltılabilir. Şekil 3.1 'de Windows Server 2003 işletimi sistemi üzerinde çalışan IIS 6.0 web sunucusunda tutulan log dosyasından örnek bir satır verilmiştir.

```
2010-03-05 00:22:31 193.140.180.4 GET
/Default.aspx - 80 - 212.154.80.164 M
ozilla/4.0+(compatible;+MSIE+6.0;+Wi
ndows+NT+5.1; +SV1;+GTB6.4) - 200 0
0 67049 428 31
```

Şekil 3.1. Log dosyalarından örnek bir satır.

Web kullanım madenciliği ön işlem, örüntü keşfi ve örüntü analizi olmak üzere 3 aşamada gerçekleştirilir [3]. Bu aşamalar Şekil 3.2'de gösterilmiştir.



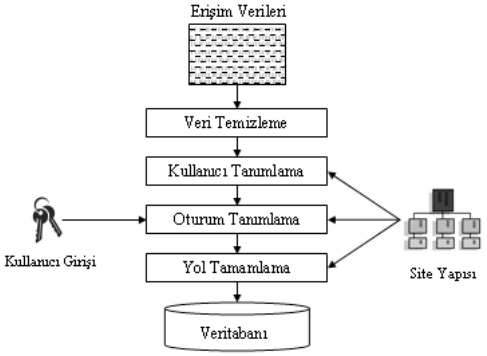
Şekil 3.2. Web kullanım madenciliğinin uygulama adımları [8].

4. Ön İşlem Süreci

Web kullanım madenciliği uygulama sürecinin en önemli aşamalarından birisi veri madenciliği ve istatistiksel algoritmaların uygulanabileceği uygun hedef veri kümesinin oluşturulmasıdır. Web sunucu üzerinde tutulan kullanıcı erişim dosyaları(log files) karmaşık, düzensiz ve herhangi bir anlam ifade etmeyecek şekilde tutulmaktadır. Web sunucusu üzerinde tutulan log dosyalarından sağlıklı bilgi çıkarımı yapabilmek için gereksiz verilerden temizlenmesi ve belirli bir düzene sokulması gerekmektedir. Sunucular üzerinde karmaşık ve düzensiz bir şekilde tutulan log dosyalarındaki verilerin analiz değeri olmayan ilişkisiz verilerden temizlenmesi, belirli bir biçime getirilmesi ve veritabanına aktarılması işlemi ön işlem sürecidir.

Ön işlem süreci web kullanım madenciliğinin en önemli ve en uzun süren basamağıdır. Bu süreç sonrasında veri örüntü keşfi için uygun hale getirilmektedir. Bu süreçte önemli olan verinin orijinalliğinin korunmasıdır.

Ön işlem süreci veri temizleme, kullanıcı tanımlama, oturum tanımlama, yol tamamlama ve biçimlendirme olmak üzere dört adımda gerçekleşir. Verilerin temizlenmesi, kullanıcı ve oturum tanımlama aşamalarında sezgisel(heuristic) teknikler kullanılmaktadır [9]. Web kullanım verisine VM tekniklerinin başarılı bir şekilde uygulanması, ön işlem sürecindeki işlemlerin doğru uygulanmasına büyük oranda bağımlıdır. Ön işlem sürecinin adımları Şekil 3.3'de gösterilmektedir.



Şekil 3.3. Web kullanım madenciliği ön işlem süreci adımları.

Veri Temizleme: Veri temizleme ön işlem sürecinde uygulanması gereken ilk adımdır. Elde edilen erişim kayıtlarının tamamı madencilik süreci için gerekli veriler değildir. Bu nedenle, erişim kayıtları içerisindeki geçerli ve gerekli olan veriler alınmalı diğerleri temizlenmelidir [11]. Temizliğe ihtiyaç duyulan gereksiz veya alakasız üç tür veri vardır. Bunlar HTML dosya içerisine gömülü ek kaynaklar, robot istekleri ve başarısız isteklerdir.

a) Ek Kaynaklar: HTTP (Hyper Text Transfer Protocol) protokolü bağlantısız bir protokol olduğu için bir kullanıcının sayfa görüntüleme isteği erişim kayıtlarında birden fazla yer alacaktır. Bunun nedeni, sayfa içerisinde kullanılan resim dosyaları, stil (css) dosyaları, script dosyaları ve sayfa içerisinde kullanılan diğer

dosyaların da erişim kayıtları içerisinde ayrı satırlar halinde yer almasıdır. Erişim kayıtları içerisinde yer alan bu tür satırlar silinmelidir.

b) Robot İstekleri: Web robotları (spider-crawler) web sitesi içerisindeki linkleri otomatik olarak çıkaran yazılımlardır. Google gibi arama motorları bir web sitesine ait tüm sayfaları ve linkleri tespit etmek için periyodik olarak bu tür araçları kullanılır. Bu tür araçlar tarafından yapılan sayfa istekleri de kullanıcı isteğinde olduğu gibi erişim kayıtları içerisinde yer alacaktır. Erişim kayıtları içerisinde yer alan bu tür kayıtlar da temizlenmelidir.

c) Başarısız İstekler: Erişim kayıtları içerisindeki her bir istek için durum kodu (sc-status) tutulmaktadır. Bu durum kodu isteğin başarılı olup olmadığını tutmaktadır. Başarısız istekler madencilik süreci için gereksiz olabilir. 200 ile 299 arasındaki durum kodları başarılı istekler olduğu için istenilirse bunlar dışında kalan istekler silinebilir. Örneğin, 404 durum kodu istekte bulunulan kaynağın var olmadığını göstermektedir. Erişim kayıtları içerisinde yer alan başarısız istekler istenilirse silinebilir. Ancak, hatalı istekler, kırık linkler veya engelli girişler gibi analiz işlemleri yapılacaksa durum kodları dikkate alınacağı için başarısız erişimler silinmemelidir.

Kullanıcı Tanımlama: Web kullanım madenciliği analizi için bir kullanıcının doğrulanmasına ihtiyaç yoktur. Fakat farklı kullanıcıları ayırt etmeye ihtiyaç duyulur.

Oturum Tanımlama: Bir oturum kullanıcının siteye girişi ile çıkışı arasındaki sürede gerçekleştirdiği aktiviteler grubu olarak tanımlanabilir. Bu nedenle oturum tanımlama işlemi, web oturumları içerisindeki her bir kullanıcının davranış ve aktivite kayıtlarının kümelenmesidir [9]. Oturum tanımlamadaki amaç oturumlar içerisindeki her kullanıcının sayfa erişimlerini birbirinden ayırt etmektir.

Yol Tamamlama: Erişim kayıtları vekil sunucuda tutuluyorsa veya site gezintisi esnasında ön bellekten sayfa ziyaretleri gerçekleşiyorsa log dosyaları içerisine kaydedilmeyen önemli erişimler vardır. Yol tamamlamanın görevi erişim kayıtları içerisinde bulunan bu eksik referansları tamamlamaktır [10].

3.2. Örüntü Keşfi

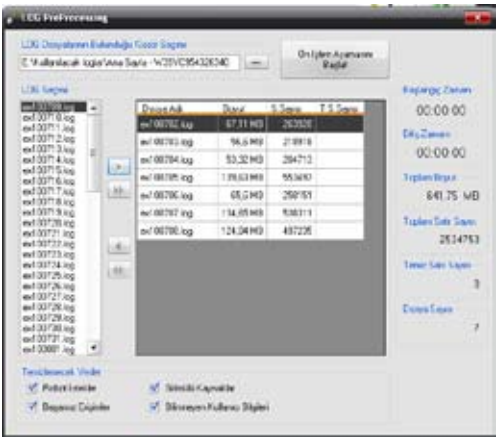
Örüntü keşfi aşamasında ön işlem sürecinden sonra elde edilen düzenli ama anlamsız olan verilerden, veri madenciliği yöntemlerini kullanarak istenilen faydalı ve gerekli bilgilerin ortaya çıkarılması gerçekleştirilmektedir.

3.3. Örüntü Analizi

Örüntü analizi web kullanım madenciliğinin son adımıdır. Örüntü analizinin amacı bulunan örüntülerden ilginç olmayan kuralları, istatistikî bilgileri ya da örüntüleri elemektir [6, 8]. Genellikle örüntü analiz işlemi web madenciliği uygulamaları tarafından elde edilir. SQL, MySQL gibi veritabanı uygulamaları ve On-Line Analytical Processing (OLAP) yaygın olarak kullanılan bilgi sorgulama mekanizmalarıdır.

4. LOG PreProcessing

LOG PreProcessing yazılımı C# kullanılarak Visual Studio 2005 ortamında geliştirilmiş ve veritabanı olarak SQL Server 2005 Express Edition kullanılmıştır.



Şekil 4.1. LOG PreProcessing ekran görüntüsü

Hazırlanan yazılım ön işlem aşamasının veri temizleme aşamasını gerçekleştirerek metin dosyalarında tutulan erişim kayıtlarını veritabanı ortamına aktarmaktadır. Şekil 4.1'de programa ait ekran görüntüsü verilmiştir.

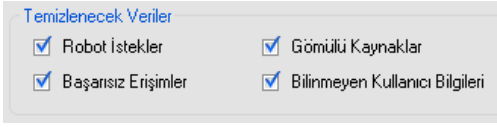
Yazılım, formlar yardımıyla tasarlanarak kullanıcının komut kullanmasına gerek kalmadan işlemleri gerçekleştirmesi sağlanmıştır.

Web kullanım madenciliği için kullanılacak erişim kayıtlarının bulunduğu klasör seçildiğinde bu klasör içerisinde bulunan log uzantılı dosyaların tamamı listeye eklenecektir. Kullanıcı dosyaların tamamını veya istediklerini listeden datagrid nesnesine ekleyebilir.

Liste içerisinden seçilip datagride eklenen dosyaların boyutları ve içerdiği satır sayıları yine datagrid içerisinde görüntülenecektir.

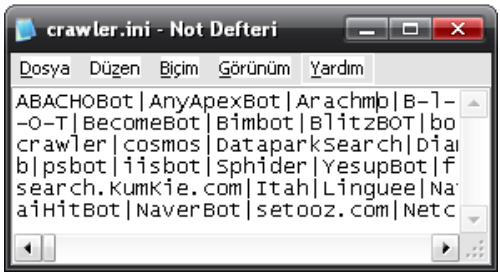
Formun sağ kısmında bulunan başlangıç zamanı ve bitiş zamanı veri temizleme işleminin süresini, toplam boyut datagrid içerisine eklenen log dosyalarının toplam boyutunu, toplam satır sayısı log dosyalarının içerdiği toplam satır sayısını ve dosya sayısı ise datagrid içerisine eklenen dosya sayısını göstermektedir. Temiz satır sayısı başlangıçta sıfır değerini içerir ama ön işlem aşaması sonrasında elde edilen temiz satır sayısını gösterecektir.

Veri temizleme işlemi yapılırken ne tür verilerin temizleneceğine Şekil 4.2'de verilen ve formun altında bulunan temizlenecek veriler kısmından seçim yapılarak karar verilebilir.



Şekil 4.2. Temizlenecek veri türleri.

Robot İstekler: Erişim kayıtları içerisinde kullanıcı erişimlerinin yanı sıra örümcek yazılımlar tarafından yapılan sayfa ziyaretleri de bulunmaktadır. Bu tür erişimlerin temizlenmesi için kullanılır. Yapılan erişimlerin robot istek olup olmadığına karar vermek için Şekil 4.3'de verilen "crawler.ini" isminde bir metin dosyası oluşturarak anahtar kelimeler eklenmiştir. Erişim satırının user-agent bilgisi bu anahtar kelimelerden birisini içeriyorsa robot istek olarak kabul edilmektedir.



Şekil 4.3. crawler.ini dosyası içeriği.

Gömülü Kaynaklar: Erişim kayıtları içerisinde bağlantılı kurulan sayfayla birlikte sayfaya ait gömülü kaynaklar da tutulmaktadır. Bu tür erişimleri temizlemek için bu seçenek seçili olmalıdır. Bu işlem gerçekleştirilirken "dosya_uzantileri.ini" isminde bir metin dosyası oluşturulmuş ve dikkate alınacak uzantılar bu dosya içerisinde belirtilmiştir. İstenilirse yeni uzantılar bu dosya içerisine eklenebilir.

Başarısız Erişimler: Erişim kayıtları içerisinde her bir isteğe ait durum kodu tutulmaktadır. Bu durum kodu isteğin başarıyla gerçekleşip gerçekleşmediğini tutmaktadır. Başarılı erişimler için 200 durum kodu kullanılmakta başarısız erişimler için hata durumuna göre kod değişmektedir. Eğer başarısız erişimler temizlenmek isteniyorsa 200 durum kodu haricinde durum koduna sahip olan erişimler temizlenecektir.

Bilinmeyen Kullanıcı Bilgileri: Erişim kayıtları içerisindeki user-agent bilgisi kullanılarak ziyaretçiye ait kullandığı tarayıcı ve işletim sistemi gibi çeşitli bilgiler elde edilebilir. Kullandığı tarayıcı veya işletim sistemi tespit edilemeyen kullanıcıların erişimlerini temizlemek için kullanılır.

4.1. Erişim Kayıtlarının Temizlenmesi ve Veritabanına Aktarılması

Erişim kayıtlarının içerdiği verilerin tamamı madencilik süreci için gerekli veriler değildir. Bu nedenle, erişim kayıtları içerisindeki geçerli ve gerekli olan veriler alınmalı diğerleri temizlenmelidir [11].

Yazılım ile temizlenecek veriler kullanıcının tercihine sunulmakta ve kullanıcının seçmiş olduğu veriler temizlenmektedir.

Yazılım, kullanıcının seçerek datagrid içerisine eklediği her bir dosyayı sırayla açıp satır satır okumaktadır. Okunan satırlar temizlenecek veri içeriyorsa atlanarak bir sonraki satırdan devam etmektedir. Elde kalan veriler biçimlendirilerek veritabanına aktarılmaktadır. Tablo 4.1'de temiz verilerin aktarılacağı tablonun içerdiği sütunlar ve veri tipleri verilmiştir.

Sütun Adı	Veri Tipi
no	bigint
tarih	datetime
saat	time
url	nvarchar(50)
referans	nvarchar(max)
status	int
bant_gens	bigint
browser	nvarchar(50)
platform	nvarchar(50)
ipcode	bigint

Tablo 4.1. log_data tablosu ve içerdiği sütunlar.

"log_data" tablosunun içerdiği sütunların tutacağı veriler aşağıda açıklanmıştır.

no: Eklenen her bir kayıt için sıra numarası vermek için kullanılır ve otomatik artan özelliğe sahiptir.

tarih: Erişim yapılan tarihi tutmaktadır.

saat: Erişim yapılan saati tutmaktadır.

url: Web sitesine ait erişim yapılan sayfayı tutmaktadır.

referans: Ziyaretçinin ziyaret ettiği sayfaya hangi kaynaktan geldiğini göstermektedir.

status: Yapılan erişim başarılı olup olmadığa ait durum kodu bilgisini tutmaktadır.

bant_gens: Erişm yapılan sayfa için kullanılan veri paketi boyutunu gösterir.

browser: Ziyaretçinin kullanmış olduğu tarayıcı bilgisini tutmaktadır.

platform: Ziyaretçinin kullanmış olduğu işletim sistemi bilgisini tutar.

ipcode: Ziyaretçinin IP adresinin sayısal karşılığı tutmaktadır. Sayısal değer kullanılmasının nedeni kullanıcının ülkesini tespit ederken kolaylık sağlaması içindir. IP adresini sayısal değere dönüştürmek için kullanılan kod bloğu Şekil 4.4'de verilmiştir.

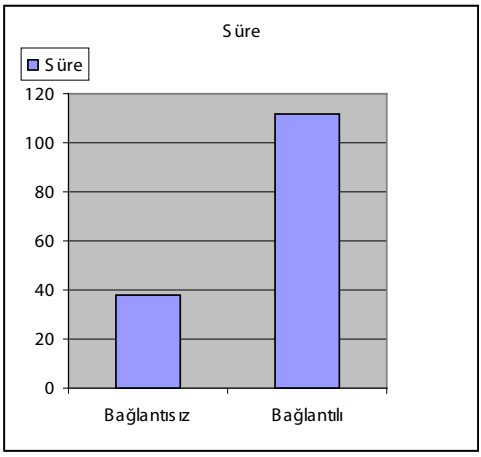
```
String ip_adresi="193.140.80.2";
String[] ip= ip_adresi.Split('.');
Double ipcode = 16777216 * Convert.
ToDouble(ip0]) + 65536 * Convert.
ToDouble(ip[1]) + 256 * Convert.
ToDouble(ip[2]) + Convert.
ToDouble(ip[3]);
```

Şekil 4.4. IP adresini sayısal değer dönüştürmek için kullanılan kod bloğu.

Temizlenen veriler veritabanına aktarılmadan önce Datatable üzerinde depolanmaktadır. Tüm dosyaların temizlik süreci sona erdikten sonra sqlbulkcopy yardımıyla datatable içerisindeki veriler veritabanına aktarılmaktadır. Datatable kullanmadan temizlik aşamasında her bir satır veritabanına aktarılmak istendiğinde veritabanına temiz satır sayısı kadar bağlantı kurmakta ve performans düşüşüne neden olmaktadır. Her iki yöntem de 198738 satır içeren, 48.3MB boyutunda 77 dosyalık erişim kayıtları üzerinde test edilmiş ve Şekil 4.5'de verilen grafik elde edilmiştir.

Grafik sonuçlarına göre bağlantısız yöntem olan datatable yardımıyla veritabanına aktarım

daha kısa sürede gerçekleştiği için bu yöntem kullanılmıştır.



Şekil 4.5. Veritabanına aktarım için geçen süreler.

Veri temizleme sonrası veritabanına aktarılan kayıtlardan örnek bir kesit Şekil 4.6'da verilmiştir.

tarih	saat	url	referans	status	bant...	browser
2010-07-02...	06:20:23...	/default.aspx	http://www.goo...	200	0	Internet E...
2010-07-02...	06:20:23...	/default.aspx	http://www.goo...	200	661	Internet E...
2010-07-02...	06:20:29...	/index.asp	-	200	2862	Internet E...
2010-07-02...	06:20:39...	/default.aspx	http://www.goo...	200	661	Internet E...
2010-07-02...	06:20:40...	/index.asp	-	200	2007	Internet E...

Şekil 4.6. Veritabanına aktarılan kayıtlardan örnek bir kesit.

4.2. Kullanıcı Tanımlama

Web kullanım madenciliği için bir kullanıcının doğrulanmasına ihtiyaç yoktur. Fakat farklı kullanıcıları ayırt etmeye ihtiyaç duyulur.

Kimlik doğrulama veya kullanıcı taraflı çerezler olmaksızın kullanıcıları tanımlamak için IP adresi ile birlikte tarayıcı ve işletim sistemi bilgilerini tutan user-agent bilgisi de kullanılır.

Kullanıcı tanımlama işlemi için ziyaretçinin IP adresi, kullandığı işletim sistemi ve tarayıcı bilgileri kullanılmaktadır. Bu üç bilgisi aynı olan erişimler tek bir kullanıcı olarak tanımlanmaktadır.

Kullanıcı tanımlama işlemi için veri temizleme sonrası elde edilen veritabanı kullanılacaktır. Bu işlem programlama tarafında gerçekleştirildiğinde veritabanına birçok kez bağlantı kurmak gerekmektedir ve bu durum ciddi performans düşüşlerine neden olacaktır. Bu nedenle hazırlanan yazılım üzerinde kullanıcı tanımlamak için doğrudan bir seçenek bulunmamaktadır.

Kullanıcı tanımlama için veritabanı üzerinde *user_create* isminde bir saklı yordam tanımlanarak kullanıcı tanımlama işlemi bu yordam yardımıyla gerçekleştirilmektedir. Yordamı oluşturmak için kullanılan T-SQL ifadesi Şekil 4.7'de verilmiştir.

```
CREATE PROCEDURE user_create
AS
INSERT INTO user_list SELECT
ipcode,browser,platform FROM
log_data GROUP BY ipcode,browser,
platform
```

Şekil 4.7. user_create yordamı.

"user_create" yordamı log_data tablosu içerisindeki kayıtları ipcode, browser ve platform sütununa göre gruplandırılarak kullanıcıları bulmakta ve bulduğu kullanıcıları user_list tablosuna eklemektedir. "user_list" tablosunun içerdiği sütunlar Tablo 4.2.'de verilmiştir.

Sütun Adı	Veri Tipi
kno	bigint
ip	nvarchar(20)
browser	nvarchar(50)
platform	nvarchar(50)

Tablo 4.2. user_list tablosu ve içerdiği sütunlar.

5. Sonuç

İnternet kullanımının her geçen gün artması web sitelerinin artmasına ve doğal olarak da sunucular üzerinde tutulan verilerin artmasına neden olmaktadır. Ziyaretçilerin site üzerindeki tüm hareketleri sunucu log dosyalarını kaydedilmektedir. Bu kaydedilen verilerin analiz edilerek yararlı bilgi haline getirilmesi web maden-

ciliği olarak geçmektedir. Birçok kurum veya kuruluş sahibi olduğu siteyi sadece şekil yönünden incelemekte ve ziyaretçilerin site üzerindeki davranışlarını dikkate almamaktadır.

Bu çalışma da, web sitesi erişim kayıtlarının daha kolay analiz edilmesini sağlamak için log dosyalarını temizleyerek veritabanına aktaran LOG PreProcessing isminde bir yazılım hazırlanmıştır. Web kullanım madenciliğinin en önemli ve uzun süren aşaması olan ön işlem süreci bu yazılım yardımıyla gerçekleştirildikten sonra standart SQL ifadeleri yardımıyla da siteye ait istatistiki bilgiler elde edilebilir.

Yazılan program web madenciliği için temel nitelikte olup geliştirilmeye uygun olarak hazırlanmıştır. Veriler temizlenerek veritabanına aktarıldığı için programa yapılacak küçük eklemelerle tarayıcı dağılımı, trafik dağılımı, ziyaret derinliği, kullanıcı ve oturum tanımlama gibi siteye ait istatistiki bilgiler görsel olarak elde edilebilir.

Kaynaklar

[1] Gürcan, F., Köse, C., ”Web İçerik Madenciliği ve Konu sınıflandırması”, ***Akademik Bilişim 2008***, Çanakkale 18 Mart Üniversitesi, Çanakkale (2008).

[2] Etzioni, O., “The World Wide Web: Quagmire or gold mine”, ***Communications of the ACM***, 39(11):65-68, (1996).

[3] Kosala, R., Blockeel, H., “Web mining research: a survey”, ***SIGKDD: SIGKDD explorations: newsletter of the special interest group (SIG) on knowledge discovery & data mining, ACM***, 2(1): 1–15 (2000).

[4] Kantardzic M., “Data Mining:Concepts, Models, Methods and Algorithms”, ***John Wiley&Sons 2003***

[5] Belen, E., Özgür, Ç., Özakar, B., “WALA : Web Erişim Kütük Araştırmacısı”, ***9.Türkiye'de İnternet Konferansı***, İstanbul(2008).

[6] Srivastava, J., Cooley, R., Deshpande, M., Tan, P., “Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data”, ***SIGKDD Explorations***, 1(2):12-23 (2000).

[7] Srivastava, J., Desikan, P., Kumar, V., “Web Mining: Concepts, Applications and Research Directions”, ***Studies in Fuzziness and Soft Computing***, 180: 275-307 (2005).

[8] Cooley, R., Mobasher, B., Srivastava, J., “Web Mining: Information and Pattern Discovery on the World Wide Web”, ***In Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)***, USA, 558 – 567 (1997).

[9] Cooley, R., Mobasher, B., and Srivastava, J., “Data Preparation for mining World Wide Web Browsing Patterns”, ***Knowledge and Information Systems***, 1:1–27 (1999).

[10] Chaofeng, L., “Research and Development of Data Preprocessing in Web Usage Mining”, ***International Conference on Management Science and Engineering***, South-Central University for Nationalities, China (2006).

[11] Liu, H., Keselj, V., “Combined mining of Web server logs and web contents for classifying user navigation patterns and predicting users' future requests”, ***Data & Knowledge Engineering***, 61:304-330 (2007).